

本周的主要工作是实现了文件作为存储索引的基本过程。以前学习的索引大部分讲得都是理想化的，假设存在数据按照特定的索引方式在内存中构建索引，如 B 树、红黑树等。但是实际应用中，为了保证索引能长期有效通常需要持久化。因此读取数据建立索引并存储于索引文件中，即使系统重启仍然能恢复索引结构。

我们以 B 树为例，实现文件存储索引。问题看起来并不复杂，但是实际中还是有一些问题，比如索引写到文件后，如何保持 B 树的结构。这里需要在父节点中存储子节点在文件中的位置，然而文件是顺序的，而且通常是先写父节点才写子节点，所以需要计算每个节点的位置后才能真正将节点信息写到文件。为简单起见，我们将每个节点存储一行，构建好 B 树后，通过遍历计算每个节点应该在文件的第几行，由此更新父节点中子节点的位置信息。最后将更新后的节点信息一并写入文件。当增加新的节点时，由于文件系统不提供删除一行数据的函数，因此需要将原来的文件中的数据读入内存并更新，然后删除原文件，再创建一个跟原来文件同名的新文件。

B 树节点可用的数据结构很多，我们是直接用两个数组分别存储 key 和子节点，如下：

```
private T [] keys;  
private BNode<T> [] children;
```

其中 keys 的数量 = children 数量 - 1

这种实现方式比较方便，但是更直观的节点数据结构可以定义为 Pair 数组，如下

```
Pair<K, rightChild>
```

再加上最左侧的 child 就构成了一个完整的节点。

重新认真阅读了 Query Shapes of Histories，一些实现细节仍不是很清楚，需要通过一些实例才能测试代码的正确性。本周只是定义了基本的接口，下周将开始逐个实现细节。

本周另外用了两天时间修改上周的论文，论文整体已经完稿，下周再花两天左右时间来压缩，并对其中的配图重新绘制并美化。

另外，今年湖南省科技厅的项目开始申报，我想把今年的国家基金重新整理下，递交上去试试。申报截止期是 11 月 10 日，所以下周任务较重。